

**Amendments to the Drawings:**

The drawing sheets attached in connection with the above-identified application containing FIGs. 1 and 2 are being presented as replacement sheets to be substituted for the previously submitted drawing sheets containing FIGs. 1 and 2. The drawing FIGs. 1 and 2 have been amended.

The specific change which has been made to FIG. 1 is that the computer program in the host 102 that is labeled on the top-right side of the figure has been relabeled from “103” to “130”.

The specific changes which have been made to FIG. 2 are: (i) the computer program in the host 102 that is labeled at the top of the figure has been relabeled from “103” to “130”; and (ii) the computer program in the remote host 135 that is labeled in the middle of the figure has been relabeled from “103” to “130”.

**REMARKS**

**Status of Claims:**

Claims 2 and 5 are cancelled. Thus, claims 1, 3-4, and 6-21 are present for examination.

**Specification:**

The specification has been amended to correct some minor informalities. No new matter has been added.

**Drawings:**

The drawing FIGs. 1 and 2 have been amended. The specific change which has been made to FIG. 1 is that the computer program in the host 102 that is labeled on the top-right side of the figure has been relabeled from "103" to "130". The specific changes which have been made to FIG. 2 are: (i) the computer program in the host 102 that is labeled at the top of the figure has been relabeled from "103" to "130"; and (ii) the computer program in the remote host 135 that is labeled in the middle of the figure has been relabeled from "103" to "130". No new matter has been added.

**Claim Rejections:**

Claims 1-4, 6-7, 10-11, and 17-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Belkin (U.S. Patent Number 6,604,125) in view of Cianfrocca et al. (U.S. Patent Number 6,088,796) (hereinafter Cianfrocca).

Claims 12-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Belkin in view of Cianfrocca, and further in view of Gutfreund et al. (U.S. Patent Number 6,192,394) (hereinafter Gutfreund).

Claims 14-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Belkin in view of Cianfrocca, and further in view of Drumm et al. (U.S. Patent Number 6,643,683) (hereinafter Drumm).

Claims 8-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Belkin in view of Cianfrocca, and further in view of Derby et al. (U.S. Patent Number 5,426,637) (hereinafter Derby).

Claim 2 has been cancelled. With respect to claims 1, 3-4, and 6-21, as amended, the rejections are respectfully traversed.

Independent claim 1, as amended, recites a system for collaborative processing with distributed applications, comprising:

“at least one application context in which an application is executed, the application context including an application CGI for managing the application, and a communication interface on which application data is communicated as messages;

a messaging bus configured to communicate the messages for processing by the application;

at least one gateway context including a gateway CGI configured for maintaining two-way asynchronous communication between the messaging bus and a remote application through a firewall, said remote application being executed by a client to a web server, the gateway CGI being configured to maintain the two-way asynchronous communication until termination by the remote application or by the gateway CGI; and

the web server, said web server being configured to establish a socket connection with said client through said firewall in response to an HTTP request from said client, said two-way asynchronous communication between said messaging bus and said remote application occurring over said socket connection.” (Emphasis Added).

A system including the above-quoted features has at least the advantages that: (i) at least one gateway context includes a gateway CGI configured for maintaining two-way asynchronous communication between a messaging bus and a remote application through a firewall, where the remote application is being executed by a client to a web server; (ii) the gateway CGI is configured to maintain the two-way asynchronous communication until termination by the remote application or by the gateway CGI; (iii) the web server is configured to establish a socket connection with the client through the firewall in response to an HTTP request from the client; and (iv) the two-way asynchronous communication between the messaging bus and the remote application occurs over the socket connection. (Applicant's FIG. 1, references 101, 106, 108, 109, 110, 112, 114, 118, 120, and arrows denoting "two-way asynchronous communications"; Specification; page 5, lines 10-15; page 7, lines 6-13; page 9, lines 12-21; page 10, line 28 to page 11, line 5).

The present application incorporates by reference co-pending U.S. Patent Application Serial No. 09/766,439 entitled "System and Method for Maintaining Two-Way Asynchronous Notification between a Client and a Web Server" (hereinafter '439 application). (Specification; page 7, lines 6-13). Applicant has enclosed a copy of the '439 application along with this reply for the Examiner's convenience.

As shown in FIG. 2 of the '439 application, asynchronous communication allows for operations in which a CGI reads and processes client requests (steps 230, 240), and such reading and processing is decoupled from operations in which the CGI determines if there is information or requests to send to the client and operations in which the CGI sends information or requests to the client (steps 232, 242). Also, as shown in FIG. 4 of the '439 application, asynchronous communication allows for operations in which the client sends client requests or information (step 430), and such sending is decoupled from operations in which the client receives information or requests and operations in which the client processes received information or requests (steps 432, 437).

Thus, with two-way asynchronous communication, there does not have to be a cause and effect relationship in operations of the CGI between reading client requests and sending information to the client. Also, with two-way asynchronous communication, there does not have to be a cause and effect relationship in operations of the client between sending client requests and receiving information. For example, information may be sent from the CGI to the client where the information is not directly in response to a client request, and operations for reading requests and sending information by the CGI may occur concurrently. ('439 application; FIGs. 2 and 4; Specification; page 3, lines 5-7 and 12-22; page 6, lines 5-24; page 8, line 27 to page 9, line 7).

In order to more fully explain the difference between synchronous and asynchronous communication, consider the example of a telephone conversation between two parties, which is a simplified example of two-way asynchronous communication. A telephone conversation is an example of two-way asynchronous communication because both parties can talk whenever they desire to talk, and one party does not have to wait for the other party to talk before talking. Also, the two parties can talk concurrently.

Synchronous communication merely allows for opening a socket connection, sending a request, executing an action, formulating responses, and sending the responses back across the connection. In such synchronous systems, the sending of responses over the socket connection can only occur in response to requests and, thus, such system do not allow for two-way asynchronous communication.

The Examiner states that, "[t]he Belkin reference teaches at least one gateway context including a gateway CGI configured for maintaining two-way asynchronous communication between the messaging bus and a remote application (Belkin: col. 3, lines 32-34; col. 4, lines 42-44)". (Emphasis Added). The Examiner further states that, "[t]he Cianfrocca reference teaches communicating through a firewall (Cianfrocca: col. 2, lines 26-34) as well as asynchronous message oriented communications (Cianfrocca: col. 2, lines 11-25, lines 47-51)."

However, neither Belkin nor Cianfrocca, alone or in combination, disclose or suggest a system including the above-quoted features for at least the following two reasons.

First, Belkin neither discloses nor suggests at least one gateway context including a gateway CGI configured for maintaining two-way asynchronous communication between a messaging bus and a remote application, where the two-way asynchronous communication occurs over a socket connection. The term “two-way asynchronous” communication is a very important limitation in applicants’ claim. This limitation is not shown in Belkin.

More specifically, as seen in FIG. 4 of Belkin, the system of Belkin only allows for a web server to receive a request (step 404), assign a thread to service the request (step 408), invoke a service with the thread to process the request (step 410), generate and send response pages in reply to the request (step 414), and then return the thread to a thread pool (step 416). (Belkin; FIG. 4; column 8, line 48 to column 11, line 3). Therefore, the communication over a socket connection in Belkin can only be synchronous, because there must first be a request received, and then response pages are sent in reply to the request. (Belkin; FIG. 4). In the synchronous communication of Belkin, there is only a cause and effect relationship between requests and responses and, thus, the sending of information cannot be decoupled from the receiving of requests. As a result, the system of Belkin does not allow for two-way asynchronous communication over a socket connection.

The Examiner points to Belkin (col. 3, lines 32-34 and col. 4, lines 42-44) as disclosing two-way asynchronous communication. However, col. 3, lines 32-36 of Belkin merely states:

“Once the proper thread pool is determined, a thread from that thread pool is used to carry out servicing of the request. By servicing the request, it is meant that a set of code is executed to carry out the functions needed to satisfy the request.” (Belkin; column 3, lines 32-36) (Emphasis Added).

The above-quoted portion of Belkin only describes synchronous communication in which a request is received and a thread is used to service or satisfy the request. Such

synchronous communication is not two-way asynchronous communication over a socket connection. Also, col. 4, lines 41-45 of Belkin merely states:

“The server 106 is the component responsible for providing most of the functionality of the system 100. More specifically, the server 106 receives requests from the clients 102, and responds to the requests by providing response pages.” (Belkin; column 4, lines 41-45) (Emphasis Added).

Again, the above-cited portion of Belkin only describes synchronous communication in which a server receives requests from clients and responds to the requests by providing response pages. Such synchronous communication is not two-way asynchronous communication over a socket connection.

Second, Cianfrocca does not cure the deficiency with respect to the teaching of Belkin, because Cianfrocca does not disclose or suggest a system in which two-way asynchronous communication occurs between a messaging bus and a remote application over a socket connection, where the remote application is being executed by a client to a web server, and where the web server establishes the socket connection with the client through a firewall in response to an HTTP request from the client.

The system of Cianfrocca includes a messenger system that is a multi-protocol server that supports HTTP and a native messenger system protocol. (Cianfrocca; column 3, line 66 to column 4, line 2). An example of a native messenger system protocol in the system of Cianfrocca is the native Tempest Messenger Protocol (TMSP). (Cianfrocca; column 4, lines 2-3). The messenger system in the system of Cianfrocca identifies a protocol that is used for a connection and treats it appropriately. (Cianfrocca; column 4, lines 7-10).

In the case that HTTP 1.0 is the protocol used to connect to the messenger system in the system of Cianfrocca, the messenger system closes the socket connection once information is sent back across the connection. (Cianfrocca; column 4, lines 11-19). Thus, any socket connection established with a client in response to an HTTP request from the client in the system of Cianfrocca is only a synchronous socket connection, because the socket connection is

closed once information is sent back across the connection. Indeed, Cianfrocca states that, “[t]he nature of a HTTP request is that there is request for connection which is made to respond to a single query.” (Cianfrocca; column 8, lines 48-50) (Emphasis Added).

In contrast, a system for collaborative processing as recited in the present claim allows for a web server to establish a socket connection with a client through a firewall in response to an HTTP request from the client, where the socket connection allows for two-way asynchronous communication between a messaging bus and a remote application. Since the system of Cianfrocca only allows for establishing a synchronous socket connection between a messaging system and a client in response to an HTTP request from the client, Cianfrocca does not disclose or suggest the features of the present claim.

While a messaging system in the system of Cianfrocca does allow for warm socket connections that are full-duplex and that remain open, such warm socket connections in the system of Cianfrocca can only be established by the messaging system in response to a Connect function that is transmitted in a native messenger system protocol. (Cianfrocca; column 4, lines 32-39). The warm socket connections in the system of Cianfrocca cannot be established between a messaging system and a client in response to an HTTP request from the client. Instead, the system of Cianfrocca requires the use of a native messenger system protocol, such as the native Tempest Messenger Protocol (TMSP). (Cianfrocca; column 4, lines 2-3 and lines 20-47).

In the system of Cianfrocca, a messenger system enabled application uses a supporting library to connect to the messenger system using the native messenger system protocol. (Cianfrocca; column 4, lines 20-22). Each connection in the system of Cianfrocca is initiated by the application itself and is terminated by the application itself, and no control over the connection arrangement exists in the messenger system. (Cianfrocca; column 5, lines 34-38). Thus, a warm socket connection in the system of Cianfrocca between a messenger system and a messenger system enabled application can only be established in response to a Connect function



transmitted from the messenger system enabled application in a native messenger system protocol, such as TMSP, and cannot be established in response to an HTTP request from the messenger system enabled application. (Cianfrocca; column 4, lines 20-47; column 6, lines 50-51).

Therefore, independent claim 1, as amended, is neither disclosed nor suggested by the Belkin and Cianfrocca references and, hence, is believed to be allowable.

Independent claim 21, as amended, recites a system with features similar to features of a system of independent claim 1 and, thus, is believed to be allowable for at least the same reasons that independent claim 1 is believed to be allowable.

The dependent claims are deemed allowable for at least the same reasons indicated above with regard to the independent claims from which they depend. Also, with respect to claims 12-13, Gutfreund does not cure the deficiency discussed above with regard to the Belkin and Cianfrocca references. With respect to claims 14-16, Drumm does not cure the deficiency discussed above with regard to the Belkin and Cianfrocca references. With respect to claims 8-9, Derby does not cure the deficiency discussed above with regard to the Belkin and Cianfrocca references.

**Conclusion:**

Applicants believe that the present application is now in condition for allowance. Favorable reconsideration of the application as amended is respectfully requested.

The Examiner is invited to contact the undersigned by telephone if it is felt that a telephone interview would advance the prosecution of the present application.

The Commissioner is hereby authorized to charge any additional fees which may be required regarding this application under 37 C.F.R. §§ 1.16-1.17, or credit any overpayment, to Deposit Account No. 50-0872. Should no proper payment be enclosed herewith, as by a check

being in the wrong amount, unsigned, post-dated, otherwise improper or informal or even entirely missing, the Commissioner is authorized to charge the unpaid amount to Deposit Account No. 50-0872.

If any extensions of time are needed for timely acceptance of papers submitted herewith, Applicants hereby petition for such extension under 37 C.F.R. §1.136 and authorizes payment of any such extensions fees to Deposit Account No. 50-0872.

Respectfully submitted,

Date 10-19-05

By David A. Blumenthal

FOLEY & LARDNER LLP  
Customer Number: 30542  
Telephone: (310) 975-7895  
Facsimile: (310) 557-8475

David A. Blumenthal  
Attorney for Applicant  
Registration No. 26,257